

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**  
**BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

|                            |   |                            |
|----------------------------|---|----------------------------|
| In re Application of:      | § | Examiner: Kiss, Eric B.    |
| Akm N. Islam               | § | Group Art Unit: 2192       |
| Michael C. Hulton          | § | Atty. Dkt. No.: 5681-68600 |
| W. Byron Nevins Jr.        | § |                            |
| Kenneth Ebbs               | § |                            |
| Sridatta Viswanath         | § |                            |
| Abhijit Kumar              | § |                            |
|                            | § |                            |
|                            | § |                            |
| Serial No. 10/726,447      | § |                            |
|                            | § |                            |
| Filed: December 3, 2003    | § |                            |
|                            | § |                            |
| For: System and Method for | § |                            |
| Application Deployment in  | § |                            |
| a Domain for a Cluster     | § |                            |

**APPEAL BRIEF**

**Mail Stop Appeal Brief - Patents**  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Sir/Madam:

Further to the Notice of Appeal filed July 31, 2008, Appellant presents this Appeal Brief. Appellant respectfully requests that the Board of Patent Appeals and Interferences consider this appeal.

**I. REAL PARTY IN INTEREST**

As evidenced by the assignment recorded at Reel/Frame 014778/0514, the subject application is owned by Sun Microsystems, Incorporated, a corporation organized and existing under and by virtue of the laws of the State of Delaware, and having its principal place of business at 4150 Network Circle, Santa Clara, CA 95054.

## **II. RELATED APPEALS AND INTERFERENCES**

No other appeals, interferences or judicial proceedings are known which would be related to, directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

### **III. STATUS OF CLAIMS**

Claims 1-15 are pending in the application and stand finally rejected. The rejection of claims 1-15 is being appealed. A copy of claims 1-15 is included in the Claims Appendix herein below.

#### **IV. STATUS OF AMENDMENTS**

No amendments have been submitted subsequent to the final rejection.

## **V. SUMMARY OF CLAIMED SUBJECT MATTER**

Independent claim 1 is directed to method. The method includes assembling files for an application. (See, e.g., Fig. 2; and p. 8, lines 21-22.) The method provides for compiling the files into an application on an administration server, where the administration server is coupled to a central application repository. (See, e.g., Fig. 2; p. 8, lines 1-2; and p. 9, lines 1-2.) The method calls for storing the application in the central application repository. (See, e.g., Fig. 2; and p. 8, lines 1-2.) The method recites after successfully compiling the application on the administration server, deploying the application from the central application repository onto a plurality of servers in a cluster of servers. (See, e.g., Fig. 2; and p. 8, lines 1-3 and lines 19-20.) The method entails starting the application on the plurality of servers. (See, e.g., Fig. 3; and p. 10, lines 8-10 and line 25-26.)

Independent claim 6 is directed to a system which includes files for an application. (See, e.g., Figs. 2-3; and p. 8, lines 21-22.) The system provides several servers in a cluster of servers. (See, e.g., Fig. 1; and p. 6, lines 3-7.) The system contains an administration server connected to the cluster of servers. (See, e.g., Fig. 1; and p. 6, lines 5-7.) The administration server is operable to compile the files into an application. (See, e.g., p. 8, lines 1-4.) The administration server is also operable to store the application in a central application repository connected to the administration server operable to store the application. (See, e.g., Fig. 2; and p. p. 8, lines 1-2.) The administration server is further operable to, after successfully compiling the application, deploy the application from the central application repository onto several servers in the cluster. (See, e.g., Fig. 2; and p. 8, lines 1-3 and lines 19-20.) The several servers are operable to start the application. (See, e.g., Fig. 3; and p. 10, lines 8-10 and line 25-26.)

Independent claim 11 is directed to a computer-readable storage medium storing program instructions computer-executable to implement a method. (See, e.g., Fig. 11; p. 24, line 14 through p. 25, line 2; and p. 25, lines 10-17.) The method includes assembling files for an application. (See, e.g., Fig. 2; and p. 8, lines 21-22.) The method

provides for compiling the files into an application on an administration server, where the administration server is coupled to a central application repository. (See, e.g., Fig. 2; p. 8, lines 1-2; and p. 9, lines 1-2.) The method calls for storing the application in the central application repository. (See, e.g., Fig. 2; and p. 8, lines 1-2.) The method recites after successfully compiling the application on the administration server, deploying the application from the central application repository onto a plurality of servers in a cluster of servers. (See, e.g., Fig. 2; and p. 8, lines 1-3 and lines 19-20.) The method entails starting the application on the plurality of servers. (See, e.g., Fig. 3; and p. 10, lines 8-10 and line 25-26.)

The summary above describes various examples and embodiments of the claimed subject matter; however, the claims are not necessarily limited to any of these examples and embodiments. The claims should be interpreted based on the wording of the respective claims.

## **VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL**

Claims 1-15 stand finally rejected under 35 U.S.C. § 102(e) as being as being anticipated by Fanshier et al. (U.S. Patent 7,206,817) (hereinafter “Fanshier”).



## VII. ARGUMENT

The Examiner rejected claims 1-15 under 35 U.S.C. § 102(c) as being anticipated by Fanshier et al. (U.S. Patent 7,206,817) (hereinafter “Fanshier”). Appellants traverse this rejection for at least the following reasons.

### **Claims 1, 6, and 15:**

**1. Fanshier clearly fails to anticipate assembling files for an application, and compiling them into an application on an administration server, where the administration server is coupled to a central application repository.**

Examiner cites Fanshier, column 2, lines 53-59, as teaching these aspects of Appellants’ claim 1. Final Action, p. 4. However, the first cited portion of Fanshier recounts only a deployment tool contacting a master deployer located in the administration server, to indicate to the master deployer the location (path) of an application archive; and the second cited portion recounts only the deployment tool initiating the distribution of the application to the target servers, and the master deployer of the administration server informing the target slave deployers so that they can download the application. There is no mention of *assembling files for an application, and compiling them into an application on the administration server*.

Moreover, throughout his description, Fanshier only discloses slave deployers, located at the individual managed servers, pulling the application from an application source, not *assembling* files for the application, and then *compiling* the files *into an application on the administration server*. For example, Fanshier recites “The slave deployer can pull the application from an application source, such as an archive file or application directory [column 2, lines 1-3].” Fanshier states that after a slave deployer has pulled the application into its own managed server, it explodes the archive and attempts to load it [column 2, lines 38-40]. There is no suggestion that files for an

application are *compiled into an application on the administration server*, which is clearly separate from the individual managed servers, as shown in FIG. 1 of Fanshier.

In the Response to Arguments section of the Final Action, the Examiner refers to teachings of Fanshier found at col. 2, lines 32-36, which actually states: “A deployment tool 100, such as a standalone application or an administration console, can contact a master deployer 104 in an administration server 102 and point the master deployer to the source 124 of an application archive file or directory.” Appellants note that telling the master deployer where to find the source of an application, as described in Fanshier, is not equivalent to *compiling assembled files for an application into an application on an administration server*. In fact, nowhere does Fanshier mention *compiling assembled files for an application into an application on an administration server*.

In the Advisory Action, the Examiner refers to applications in Fanshier which are archived using the .ear format. The Examiner equates Fanshier’s storage of an archived application on an administration server with compiling the assembled files for an application into an application on an administration server, as recited in claim 1. This is an erroneous notion. Simply storing an application in an archival format is not the same as *compiling assembled files for an application into an application*.

**2. Fanshier clearly fails to anticipate storing, in the central application repository, the application that resulted from said assembling and said compiling on the administration server.**

As already explained above, Fanshier does not teach *compiling assembled files for an application into an application on an administration server*, much less *storing such an application in the central application repository*. There is no description in Fanshier of storing an application that has been assembled and compiled on the administration server in a central application repository coupled to the administration server.

3. Fanshier clearly fails to anticipate, after successfully compiling the application on the administration server, deploying the application from the central application repository onto a plurality of servers in a cluster of servers.

In fact, Fanshier teaches that the application is pulled from the application source by a target slave deployer to a staging area for a target managed server, and that at the staging area, the slave deployer explodes the application in *preparation* for activation [column 2, lines 38-40]. The slave deployer then moves the files from the staging area to the deployment directories [column 2, lines 64-67] in preparation for actual deployment (activation) [column 2, lines 29-31]. Thus, Fanshier teaches the use of a separate *staging area at the target server* from which the *application is prepared for deployment on that target server*.

Moreover, Fanshier reveals that if the application is not to be staged, it may not even reside on the administration server, but rather be deployed directly at the target server from a local directory [column 4, lines 5-8]. In Fanshier, the second phase, the *actual deployment*, or “activation,” of the application occurs at each individual managed server [column 2, lines 25-31], and the slave deployers at each managed server copy files from their staging directories to the deployment directories [column 2, lines 64-67]. If the slave deployers at the managed servers report successful loading of an application, then the master deployer can direct all the individual relevant slave deployers to activate the installation [column 2, lines 43-46]. *The actual deployment occurs from deployment directories within the managed servers, not from a central application repository.* According to Fanshier, “The master deployer can contact a slave deployer on each managed server that is to deploy the application, indicating that *the slave deployer should prepare to deploy the application*” [column 1, lines 64-67, emphasis added].

In the Response to Arguments section of the Final Action, the Examiner merely refers to the same portions of Fanshier and does not provide any substantive rebuttal of Appellants’ arguments. There is no mention whatsoever in Fanshier of the application

being deployed *from a central repository after successfully compiling the application on the administration server.*

Anticipation requires the presence in a single prior art reference disclosure of *each and every element* of the claimed invention, *arranged as in the claim*. M.P.E.P 2131; *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). The *identical* invention *must be shown in as complete detail* as is contained in the claims. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). As shown above, Fanshier fails to anticipate Appellants' claimed invention for numerous reasons.

**Claims 2, 7, and 12:**

**1. Fanshier clearly fails to anticipate storing a different version of the application in the central application repository.**

The Examiner refers to Fanshier at column 6, lines 10-16, and column 8, lines 4-8, as teaching this aspect of Appellants' claim. Final Action, p. 5. However, the first cited portion of Fanshier actually describes operations at a particular managed target server, not the central application repository. Fanshier indicates that a particular target server can activate the various applications targeted to it, and that any application which is out of date can be updated by downloading from the administration server. There is no mention of storing a different version of the same application on the central application repository. The second cited portion of Fanshier at column 8, lines 4-8 discloses that it may be necessary for all servers to be using the same software version, and that in some instances, a particular target server may deploy different versions of an application. However, no mention is made of the *same central application repository* storing a different version of the application.

In the Advisory Action, the Examiner cites Fanshier at column 6, lines 10-28, in which Fanshier states that a managed server's out-of-date applications can be updated *before being activated*, and that application deployment requests missed during an outage can be processed by a managed server following the outage. Contrary to the Examiner's assertion, this has no bearing on *storing different versions of the application in the central application repository*. In the Advisory Action the Examiner also cites Fanshier at column 6, lines 34-39, in which Fanshier recites cautioning a user not to use the configured source path of the application as the application development area, in order to prevent premature updating of an application at a managed server. Again, contrary to the Examiner's assertion, this has no bearing on *storing different versions of the application in the central application repository*.

**2. Fanshier clearly fails to anticipate deploying the different version of the application from the central application repository onto a plurality of servers in the cluster of servers, where the application and the different version of the application simultaneously execute on different servers in the cluster of servers.**

The Examiner refers to Fanshier at column 6, lines 10-16, and column 8, lines 4-8, as teaching this aspect of Appellants' claim. Final Action, p. 5. However, as explained above, Fanshier makes no mention of *storing a different version of the application in the central application repository*; furthermore, there is no indication of deploying the different version of the application from the central application repository onto a plurality of servers, where the application and the different version of the application simultaneously execute on different servers in the cluster of servers.

In the Advisory Action, the Examiner cites Fanshier at column 6, lines 10-28, in which Fanshier states that a managed server's out-of-date applications can be updated *before being activated*, and that application deployment requests missed during an outage can be processed by a managed server following the outage. Contrary to the Examiner's assertion, this has no bearing on *deploying the different version of the application from the central application repository onto a plurality of servers in the cluster of servers*,

*where the application and the different version of the application simultaneously execute on different servers in the cluster of servers.*

In the Advisory Action the Examiner also cites Fanshier at column 6, lines 34-39, in which Fanshier recites cautioning a user not to use the configured source path of the application as the application development area, in order to prevent premature updating of an application at a managed server. Again, contrary to the Examiner's assertion, this has no bearing on *deploying the different version of the application from the central application repository onto a plurality of servers in the cluster of servers, where the application and the different version of the application simultaneously execute on different servers in the cluster of servers.*

**3. Fanshier clearly fails to anticipate starting an older version of the application on a subset of servers such that the older version and a current version of the application simultaneously execute on different servers in the cluster of servers.**

The Examiner refers to Fanshier at column 6, lines 10-39 as teaching this aspect of Appellants' claim. Final Action, p. 5. However, the cited portion of Fanshier only discusses outages and updates at the target servers. There is no mention at all of *starting an older version of the application on a subset of servers such that the older version and a current version of the application simultaneously execute on different servers in the cluster of servers.*

In the Advisory Action, the Examiner cites Fanshier at column 6, lines 10-28, in which Fanshier states that a managed server's out-of-date applications can be updated *before being activated*, and that application deployment requests missed during an outage can be processed by a managed server following the outage. Contrary to the Examiner's assertion, this has no bearing on *starting an older version of the application on a subset of servers such that the older version and a current version of the application simultaneously execute on different servers in the cluster of servers.*

In the Advisory Action the Examiner also cites Fanshier at column 6, lines 34-39, in which Fanshier recites cautioning a user not to use the configured source path of the application as the application development area, in order to prevent premature updating of an application at a managed server. Again, contrary to the Examiner's assertion, this has no bearing on *starting an older version of the application on a subset of servers such that the older version and a current version of the application simultaneously execute on different servers in the cluster of servers.*

### **Claims 3, 8, and 13:**

**1. Fanshier clearly fails to anticipate that the assembling includes assembling the files for an application into a single archive file.**

The Examiner refers to Fanshier at column 2, lines 32-36, as teaching this aspect of Appellants' claim. Final Action, p. 5. However, the cited text makes no mention at all of *the act of* assembling the files into a single archive file; instead, it recites a deployment tool contacting a master deployer located in the administration server, to indicate to the master deployer the location (path) of an application archive. Elsewhere Fanshier describes exploding an application archive file at the target managed server, but nowhere does he suggest *the act of* assembling the files for an application into a single archive file.

In the Advisory Action, the Examiner cites Fanshier at column 3, lines 11-30, in which Fanshier recites that an enterprise application can be archived using the .ear format, and that its various standalone modules, such as EJB, web, and connector modules, may or may not be archived [see accompanying Table 1]. Fanshier does not indicate that the standalone modules are collected into a *single* archive. In fact, Fanshier states that the standalone modules are archived into files using other formats. In particular, EJBs are archived as .jar files, and connectors are archived as .rar files. Fanshier does not teach that the files for an application are assembled into a *single archive file.*

**Claims 4, 9, and 14:**

**1. Fanshier clearly fails to anticipate that the files assembled for compilation into an application comprise a set of source code and related resource files to be compiled into the application on the administration server.**

The Examiner refers to Fanshier at column 3, lines 11-30 as teaching this aspect of Appellants' claim. Final Action, p. 5. However, the cited portion of Fanshier only describes the archiving of various types of applications, such as enterprise applications, web applications, EJB, and connectors. There is no mention whatsoever of *source code and related resource files* included in a set of files *assembled for compilation on the administration server*. In fact, throughout the entire disclosure, *Fanshier never mentions source code compiled into an application on an administration server*.

The Examiner assertion in the Advisory Action that “the application files of Fanshier inherently come from source code and related resource files” is of a very general nature, and fails to address the full limitations of the Appellants' claim. In particular, the assembled source code and related resource files recited in Appellants' claim 4 must be compiled into an application *on the administration server*. Fanshier does not teach this limitation of Appellants' claim 4.

**2. Fanshier clearly fails to anticipate validating that the compilation of the files into the application on the administration server has been performed without errors.**

The Examiner refers to Fanshier at column 2, lines 38-49, as teaching this aspect of Appellants' claim. Final Action, p. 5. However, the cited portion of Fanshier describes the slave deployer's pulling a file into a staging directory, exploding it, and loading it. The slave deployer may inform the master deployer about the success of the load, thus verifying the successful loading of an application at the target managed server. This



verification of successful loading at a target server in Fanshier is in contrast to *validating that compiling source code and resource files into an application on the administration server has been performed without errors*, as recited in Appellants' claim 4.

In the Advisory Action, the Examiner asserts that "the fact that the received archive can be exploded and loaded on the managed servers validates that the files were properly compiled into the archive on the administration server." The Examiner's reference is to applications stored in archival formats for deployment [column 3, lines 11-30]. Fanshier's recitation of applications awaiting deployment as archival files has absolutely no bearing on *validating that compiling source code and resource files into an application on the administration server has been performed without errors*, and the entire reference lacks any suggestion of this aspect of the Appellants' claim.

**3. Fanshier clearly fails to anticipate that the validating is performed before deploying the application from the central application repository to any of the plurality of servers in the cluster.**

The Examiner refers to Fanshier at column 2, lines 38-49, as teaching this aspect of Appellants' claim. Final Action, p. 5. However, the cited portion of Fanshier does not teach *validating that compiling source files into an application on the administration server has been performed without errors*, as explained above. Rather, it discloses the slave deployer's pulling a file into a staging directory, exploding it, loading it, and then informing the master deployer about the success of the load. There is no mention of *validating that compiling source files and resource files into an application on the administration server has been performed without errors, and performed before deploying the application from the central application repository* to any of the plurality of servers in the cluster, as recited in claim 4. The verification of Fanshier is only a verification that files exploded at a target managed server have been successfully loaded at the target server.

In the Advisory Action, the Examiner refers to the “load-and-validate phase, that is, the preparation phase of Fanshier, in which the slave deployer on a managed server pulls a file into a staging directory, explodes the archive, and attempts to load the archive [column 2, lines 38-43]. The “load” recited in this connection is the loading of the archive *at the managed server*. The “validation” is *the managed server’s reporting* about the success of the load *to the master deployer in the administration server*. The loading and validating in Fanshier occur *at the managed servers, not at the administration server*. The load-and-validate procedure occurring at the managed servers of Fanshier has no bearing on validating that compiling source files and resource files into an application *on the administration server* has been performed without errors, *and performed before deploying the application from the central application repository of the administration server* to any of the plurality of servers in the cluster, as recited in Appellants’ claim.

**Claims 5, 10, and 15:**

**1. Fanshier clearly fails to anticipate that deploying the different version of the application comprises stopping an instance of the application on a server.**

The Examiner refers to Fanshier at column 2, lines 38-50 and at column 6, lines 1-28, as teaching this aspect of Appellants’ claim. Final Action, p. 6. However, as explained earlier, the first cited portion of Fanshier recounts only the slave deployer’s pulling a file into a staging directory, exploding it, loading it, and then informing the master deployer about the success of the load. There is no mention of *stopping an instance* of the application on a server. The second cited portion of Fanshier outlines deployment and updating of applications during new server startup, as well as redeployments after outages. Again, there is no mention of *stopping an instance* of the application on a server.

**2. Fanshier clearly fails to anticipate deploying the different version of the application comprises loading the different version of the application on the server, associating the stopped instance of the application with the different version**

**of the application on the server, and starting the instance of the different version of the application on the server.**

The Examiner again refers to Fanshier at column 2, lines 38-50 and at column 6, lines 1-28, as teaching these aspects of Appellants' claim. Final Action, p. 6. However, neither here nor elsewhere does Fanshier teach *stopping an instance* of the application on a server, much less *loading the different version of the application on the server, associating the stopped instance of the application with the different version of the application on the server, and starting the instance of the different version of the application on the server.*

In the Advisory Action, the Examiner refers to the deployment of updates to applications recited at column 6 of Fanshier, asserting that Fanshier's deployment of updates "inherently requires deactivation of the previous version of the deployed application as it is replaced either through partial updates or full redeployment." However, the cited portion of Fanshier teaches that *during the startup process*, the managed server can download any configured applications that are older than indicated in the domain configuration *before the applications are activated*. Fanshier does not teach *stopping an instance* of the application on a server, much less *loading the different version of the application on the server, associating the stopped instance of the application with the different version of the application on the server, and starting the instance of the different version of the application on the server.*

## CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1-15 was erroneous, and reversal of his decision is respectfully requested.

The Commissioner is authorized to charge the appeal brief fee and any other fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5681-68600/RCK.

Respectfully submitted,

/Robert C. Kowert/  
Robert C. Kowert, Reg. #39,255  
Attorney for Appellants

Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C.  
P.O. Box 398  
Austin, TX 78767-0398  
(512) 853-8850

Date: September 30, 2008

## VIII. CLAIMS APPENDIX

The claims on appeal are as follows.

1. A method, comprising:

assembling files for an application;

compiling the files into an application on an administration server, wherein the administration server is coupled to a central application repository;

storing the application in the central application repository;

after successfully compiling the application on the administration server,  
deploying the application from the central application repository onto a plurality of servers in a cluster of servers; and

starting the application on the plurality of servers.

2. The method of claim 1, further comprising:

storing a different version of the application in the central application repository;

deploying the different version of the application from the central application repository onto a plurality of servers in the cluster of servers, wherein the application and the different version of the application simultaneously execute on different servers in the cluster of servers; and

starting an older version of the application on a subset of servers such that the older version and a current version of the application simultaneously execute on different servers in the cluster of servers.

3. The method of claim 1, wherein said assembling comprises assembling the files into a single archive file.

4. The method of claim 1, wherein said files comprise a set of source code and related resource files for the application, the method further comprising validating that said compiling the files into the application has been performed without errors, wherein said validating is performed prior to performing said deploying the application to any of the plurality of servers in the cluster.

5. The method of claim 2, wherein deploying the different version of the application comprises:

stopping an instance of the application on a server;

loading the different version of the application on the server;

associating the instance of the application with the different version of the application on the server;

starting the instance of the different version of the application on the server.

6. A system comprising:

files for an application;

a plurality of servers in a cluster of servers;

an administration server connected to the cluster of servers, wherein the administration server for the cluster of servers is operable to:

compile the files into an application;;

store the application in a central application repository connected to the administration server operable to store the application; and

after successfully compiling the application on the administration server,  
deploy the application from the central application repository onto  
the plurality of servers in the cluster;

wherein the plurality of servers are operable to start the application.

7. The system of claim 6,

wherein the central application repository is operable to store a different version  
of the application;

wherein the administration server is operable to deploy the different version of the  
application onto a plurality of servers in the cluster of servers; and

wherein the plurality of servers are operable to simultaneously execute the  
application and the different version of the application.

8. The system of claim 6, wherein the files are assembled in a single archive  
file.

9. The method of claim 6, wherein said files comprise a set of source code  
and related resource files for the application, and wherein the administration server is  
further operable to validate that the application has been compiled without errors prior to  
the application being deployed to any of the plurality of servers in the cluster .

10. The method of claim 7, wherein the administration server is further operable to:

stop an instance of the application on a server;

load the different version of the application on the server;

associate the instance of the application with the different version of the application on the server;

start the instance of the different version of the application on the server.

11. A computer-readable storage medium storing program instructions computer-executable to implement a method comprising:

assembling files for an application;

compiling the files into an application on an administration server, wherein the administration server is coupled to a central application repository;

storing the application in the central application repository;

after successfully compiling the application on the administration server, deploying the application from the central application repository onto a plurality of servers in a cluster of servers; and

starting the application on the plurality of servers.

12. The computer-readable storage medium of claim 11, wherein the method further comprises:



storing a different version of the application in the central application repository;

deploying the different version of the application onto a plurality of servers in the cluster of servers, wherein the application and the different version of the application simultaneously execute on different servers in the cluster of servers; and

starting an older version of the application on a subset of servers such that the older version and a current version of the application simultaneously execute on different servers in the cluster of servers.

13. The computer-readable storage medium of claim 11, wherein said assembling comprises assembling the files into a single archive file.

14. The computer-readable storage medium of claim 11, wherein said files comprise a set of source code and related resource files for the application, and wherein the method further comprises validating that said compiling the files into the application has been performed without errors, wherein said validating is performed prior to performing said deploying the application to any of the plurality of servers in the cluster.

15. The computer-readable storage medium of claim 12, wherein deploying the different version of the application comprises:

stopping an instance of the application on a server;

loading the different version of the application on the server;

associating the instance of the application with the different version of the application on the server;

starting the instance of the different version of the application on the server.

## **IX. EVIDENCE APPENDIX**

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

**X.     RELATED PROCEEDINGS APPENDIX**

There are no related proceedings.